**Описание приложения:**

Приложение — это менеджер задач с графическим интерфейсом, позволяющий отслеживать активные процессы на компьютере и управлять ими (например, завершать процессы), а также отображать графики производительности в реальном времени. Оно состоит из двух основных модулей:

**main.py** — отвечает за получение информации о процессах и запуск мониторинга.

**GUI.py** — графический интерфейс для отображения списка процессов, управления ими и визуализации производительности.

**Функциональность:**

* Получение списка активных процессов с информацией о PID, имени, объеме используемой памяти и загрузке процессора.
* Автоматическое обновление списка процессов.
* Упорядочивание списка по любому из столбцов.
* Завершение выбранного процесса.
* Вкладка с графиками загрузки CPU и использования памяти в реальном времени.

**Использованные технологии и библиотеки:**

* Python 3 — основной язык программирования приложения.
* psutil — библиотека для получения информации о системных процессах, памяти и загрузке CPU.
* tkinter — стандартная библиотека Python для создания графического интерфейса.
* threading — библиотека для запуска фонового потока мониторинга.
* time — библиотека для управления интервалами обновления информации.
* matplotlib — библиотека для построения графиков производительности.

**Классы и функции:**

**main.py**

TaskMngrInfo — класс, представляющий информацию о процессе.

get\_process\_info() — функция, собирающая информацию о процессах.

monitoring\_system() — функция, непрерывно выводящая список процессов в консоль.

start\_monitoring() — функция, запускающая поток мониторинга.

**GUI.py**

TaskManagerGUI — класс, реализующий графический интерфейс.

refresh\_process\_list() — обновление списка процессов.

setup\_performance\_charts() — настройка графиков производительности.

update\_performance\_charts() — обновление данных на графиках в реальном времени.

run\_gui() — запуск интерфейса.